DG Comics: Semi-Automatically Authoring Graph Comics for Dynamic Graphs

Joohee Kim, Hyunwook Lee, Duc M. Nguyen, Minjeong Shin, Bum Chul Kwon, Sungahn Ko*, and Niklas Elmqvist

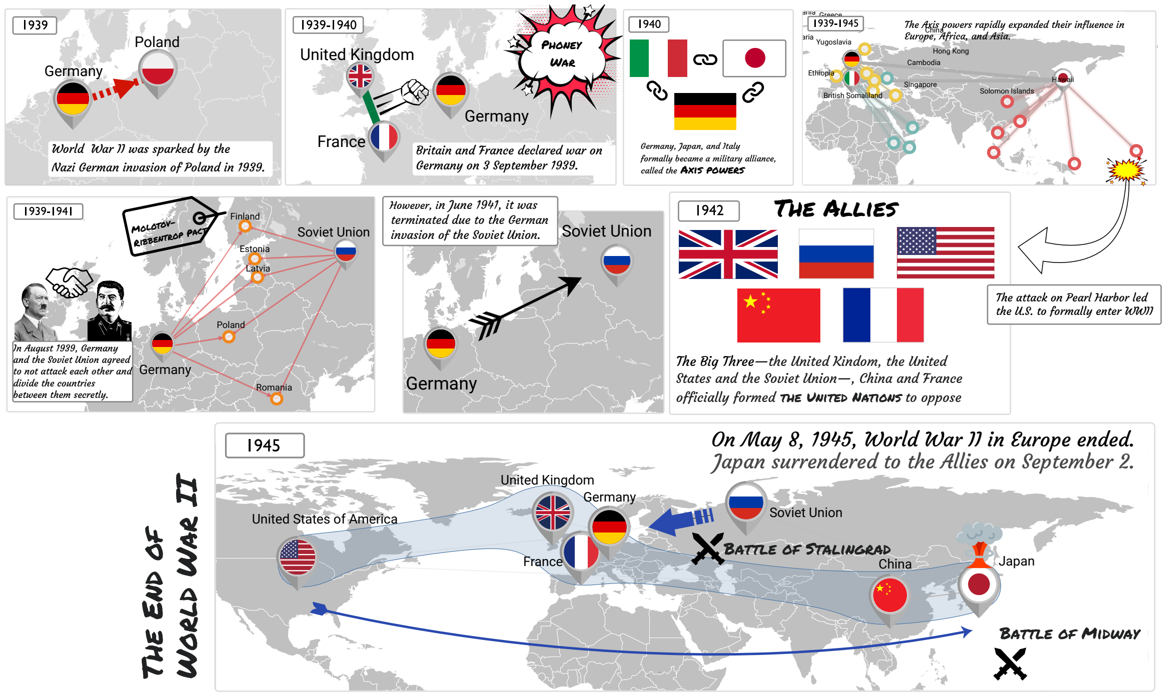

1939
Poland
Germany
World War II was sparked by the Nazi German invasion of Poland in 1939.
1939-1940
United Kingdom
Phoney War
Germany
France
Britain and France declared war on Germany on 3 September 1939.
1940
Germany, Japan, and Italy formally became a military alliance, called the Axis Powers
1939-1945
The Axis powers rapidly expanded their influence in Europe, Africa, and Asia.
Greece
China
Yugoslavia
Hong Kong
Cambodia
Ethiopia
Singapore
Hawaii
British Somaliland
Solomon Islands
1939-1941
Molotov-Ribbentrop Pact
Finland
Soviet Union
Estonia
Latvia
Poland
Germany
Romania
In August 1939, Germany and the Soviet Union agreed to not attack each other and divide the countries between them secretly.
However, in June 1941, it was terminated due to the German invasion of the Soviet Union.
Soviet Union
Germany
1942
The Allies
The Big Three—the United Kindom, the United States and the Soviet Union—, China and France officially formed The United Nations to oppose
The attack on Pearl Harbor led the U.S. to formally enter WWII
The End of World War II
1945
On May 8, 1945, World War II in Europe ended.
Japan surrendered to the Allies on September 2.
United Kingdom
Germany
United States of America
Soviet Union
France
Battle of Stalingrad
China
Japan
Battle of Midway

Fig. 1: **World War II.** Graph comic on the history of World War II, covering the causes and significant events, such as the formation of alliances and major battles, created using DG Comics. Note that while the underlying dynamic graph comic was generated automatically, the DG Comics tool provides functionality for the designer to manually move nodes, add visual highlighting, and insert the geographic map backgrounds. We used contemporary national flags for easier recognition wherever feasible.

Abstract—Comics are an effective method for sequential data-driven storytelling, especially for dynamic graphs—graphs whose vertices and edges change over time. However, manually creating such comics is currently time-consuming, complex, and error-prone. In this paper, we propose DG Comics, a novel comic authoring tool for dynamic graphs that allows users to semi-automatically build and annotate comics. The tool uses a newly developed hierarchical clustering algorithm to segment consecutive snapshots of dynamic graphs while preserving their chronological order. It also presents rich information on both individuals and communities extracted from dynamic graphs in multiple views, where users can explore dynamic graphs and choose what to tell in comics. For evaluation, we provide an example and report the results of a user study and an expert review.

Index Terms—Data-driven storytelling, narrative visualization, dynamic graphs, graph comics.

1 Introduction

Dynamic systems are prevalent in both nature and society. Catalysts facilitate chemical reactions, while species interact throughout evolution. Scientists collaborate with colleagues and students across various stages of their careers. Social relationships form, evolve, and dissolve as individuals make friends, have children, and experience rifts or pass away. Modeling these phenomena as *dynamic graphs*, where nodes represent entities and links represent their evolving relationships over time, is a valuable tool for understanding such systems. However, due to inherent complexity and scale, it is challenging to communicate stories extracted from dynamic graphs succinctly and accurately. In previous studies, *graph comics*, a comic-based storytelling medium that consists of graph visualizations, is found to be effective in narrating stories involving dynamic graphs [4, 65, 86]. Despite the potential benefits, it is time-consuming for users to create such graph comics based on their data, especially given the lack of dedicated authoring tools.

We address this gap by proposing an approach to automatically generate graph comics from hierarchical clustering that preserves the temporal causality (thus *causality-preserving*) of the events. To validate

*Corresponding author
- Joohee Kim, Hyunwook Lee, Duc M. Nguyen, and Sungahn Ko are with UNIST (Ulsan National Institute of Science and Technology). E-mail: {joohee, gusdnr0916, ducnm, sako}@unist.ac.kr
- Bum Chul Kwon is with IBM Research. E-mail: bumchul.kwon@us.ibm.com
- Niklas Elmqvist is with Aarhus University. E-mail: elm@cs.au.dk

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

the approach, we implement an interactive tool called DG COMICS:

- T1 **Storytelling Automation:** To combat the complexity of dynamic graphs with many relationships spanning a long period of time, DG Comics automates the clustering of temporal events into segments using causality-preserving hierarchical aggregation [24].
- T2 **Storytelling Agency:** To facilitate the creative and functional agency of the analyst authoring the story, DG Comics yields control of the aggregation level, the main and supporting characters to display, as well as style and formatting choices to the analyst.

DG Comics offers a causality-preserving clustering of graph snapshots in a dendrogram, which users can leverage to initiate graph comic generation. From a chosen aggregation level, which determines the number of panels, DG Comics automatically creates a comic template with a computed layout. Each panel features the main character(s) undergoing the most significant changes and their relationships. These changes are depicted with distinct visual representations and captions generated using a template-based approach. Users can further develop the template using various editing functions and build original stories through interaction with additional views and tables. We verified DG Comics's usefulness and versatility through (1) an example using a VIS coauthorship network, (2) a user study involving 13 university participants designing graph comics with an international trade network dataset, and (3) interviews with six experts from various domains.

The contributions of this work include (1) robust design requirements derived from challenges identified in prior research; (2) development of a causality-preserving clustering technique with enhanced graph similarity computation as the backbone of automation; (3) implementation of a novel interactive comic authoring tool; and (4) results from an example scenario, a user study, and expert reviews.

2 RELATED WORK

Our work lies at the intersection of graph visualization [75], data-driven storytelling [59], and data comics [86]. Below we review the literature in all these areas and discuss how our work supersedes prior art.

2.1 Graph Visualization

A *graph* or a *network* consists of nodes and links, with nodes representing entities (e.g., people, organizations) and links representing relationships (e.g., friendships, alliances). Graphs are widely used in domains [75] such as transportation, biology, communication, business, and security. As their utility increases, so does their size. Combining networks with data like maps (geospatial graphs), time (dynamic graphs) [25], or text adds complexity. Conventional graph visualizations include two methods: node-link diagrams [28], which use lines to connect nodes and visual channels (e.g., color, size, line thickness) to distinguish attributes, and adjacency matrices [12, 30, 31], which organize nodes in a grid and display connections at intersecting cells. Prior art [51, 55, 75] provides detailed descriptions.

As graphs grow, visualizations become complex and cluttered. NetworkNarratives [46] provides semi-automated guided data tours to facilitate the navigation of complex networks. Symbolic representations are one way to overcome this issue by aggregating or hiding nodes. Dunne and Shneiderman [23] propose a simplification technique that uses fan, connector, and clique motifs to save space and improve understanding of large graphs. To prevent misunderstanding and address the complexity of large graphs, some research [30, 31] combines node-link diagrams and adjacency matrices, enabling efficient exploration of networks. Yoghourdjian et al. [83] introduce graph thumbnails for high-level structure visualization, allowing easy identification, comparison, and overview of multiple large graphs. Ghani et al. [28] use dynamic insets to show off-screen node neighborhoods, while May et al. [49] improve off-screen awareness by providing graph neighbor information. Visualizing groups or clusters in graphs is another area of research. Saket et al. [63] propose a taxonomy for graph groups, enumerating tasks such as group-only, group-node, group-link, and group-network. Vehlow et al. [74] survey techniques for visually presenting graph groups, categorizing them into visual node attributes, juxtaposed, superimposed, and embedded methods.

2.2 Visualizing Dynamic Graphs

Dynamic graphs are graphs whose relations among entities change over time. Such changes bring challenges in tasks, visualization, and evaluation, prompting significant research efforts. Ahn et al. [1] categorize temporal features of dynamic graphs by the rate and shape of changes and individual events. Beck et al. [11] survey dynamic graph visualization techniques, focusing on presentation methods such as animated diagrams and static timeline-based charts. Analyses of temporal features from individual node/link, group, and network perspectives inspired how DG Comics presents information on dynamic graphs. Numerous techniques have been developed for visualizing dynamic graphs. Elzen et al. [71] visualize graph evolution with sequence views, and Burch et al. [16] propose pixel-oriented visualizations. GeneaQuilts [13] use a hybrid adjacency matrix and node-link representation to visualize family trees over time. Bach et al. [3, 5] employ 3D matrix cubes and small multiples of adjacency matrices.

To reduce the complexity of dynamic graphs, computational methods like spectral graph wavelets [22] and diachronic node embedding [81] hierarchically aggregate [24] graph snapshots based on graph structures [2] or attributes [29]. The computation results effectively visualize the changes by time (e.g., small multiples [3]). For example, Elzen et al. [72] present a novel visual analytics pipeline that allows users to track graph changes with points. The pipeline consists of democratization, vectorization and normalization, dimensional reduction, and visualization, transforming graph snapshots into points. Cakmak et al. [18] propose multi-scale snapshot visualization with graph2vec [53], creating temporal summaries of dynamics graphs. We refer to [80] for an extensive survey result on graph learning algorithms.

2.3 Graph Comics and Authoring Tools

Comics [50] are a storytelling genre that presents stories with combinations of illustrations, text, and annotations on various layouts.Segel and Heer propose using comics for data-driven storytelling in their seminal work on narrative visualization [65]. Data comics, an emerging medium of data-driven narrative visualization, leverage the visual language of comics, including layouts, characters, and captions [6,86]. Research has focused on identifying characteristics of data comics [70], developing authoring tools [20, 86], and specific applications such as visualization education [76] and user study reports [77]. Bach et al. [4] conduct a design study on storytelling with dynamic networks, proposing design factors: visual representation of graph elements and changes, temporality of changes, element identity, cast of characters, level of detail, overview and detail, and representation of multivariate networks. They also define a design space for data comics by analyzing common patterns in existing storytelling media (e.g., infographics, data videos) [7]. This design space has two dimensions: content relation patterns (e.g., narrative, temporal, faceting, visual encoding, granular, spatial patterns) and panel layout (e.g., linear, tiled, parallel, grid). Wang et al. [79] conduct controlled and in-the-wild user studies to investigate the benefits of data comics compared to infographics. Their experiments reveal that data comics improve understanding and recall of information, and are preferred for enjoyment, focus, and engagement.

Several data comics authoring tools have been proposed for various computing environments (e.g., tablets, computational notebooks, coding environments). DataToon [40] is the first effort to help users design comics on dynamic graphs with pen and touch interaction. Computational notebooks are a new tool for analyzing, visualizing, and sharing datasets. However, their results, which combine programming code, notes, and analysis, often struggle to communicate with the audience. To address this, Kang et al. [38] propose ToonNote, an extension that converts notebooks into data comics. CodeToon [69], similar to ToonNote in motivation, focuses more on coding environments, facilitating code-aligned storytelling and automated comic generation. Most data comics are static to guide readers through a specific flow and layout. Wang et al. [78], posing questions on the interactivity of data comics, formalize operations for interactive comics (e.g., content highlighting, panel addition/removal). They also present a lightweight scripting approach with six goals: navigation, details on demand, changing perspective, branching, pause and reveal, and input data.

2.4 Comparison to Prior Art

Our proposed work in this paper uses comics to visualize dynamic graphs changing over time, and our approach is novel over the literature:

- Building on Segel and Heer [65]'s taxonomy, Zhao et al. [86] first proposed data comics for narrative visualization, but our approach goes beyond their work by applying the idea to graphs.
- While Bach et al. [4] apply comics to graphs, our work proposes automatic graph comic generation as well as an interactive editor.
- Authoring tools for data comics exist [38, 40, 69, 86], but ours is unique to dynamic graphs and semi-automatically builds comics.
- Prior work visualizes temporal summaries of dynamic graphs [3, 18, 85], but we apply the idea to automating data comics.

3 Challenges and Design Requirements

The aim of this paper is to develop an authoring tool for data comics that automatically generates a narrative in comic form and enables users to efficiently reorganize them into a coherent storyline. We accomplish this by adhering to the two design principles outlined in Sec. 1: *storytelling automation* versus *storytelling agency*. These principles guide authors in managing scale and complexity while preserving their creative and expressive vision for the data-driven story.

To achieve these goals, we reviewed existing fundamental design principles formalized for storytelling [7, 33, 42, 45, 65, 68] and data comics [4, 6, 7, 84, 86]. We also based our work on prior research in dynamic graphs regarding visual analysis [37, 51, 74, 75]), tasks [1, 11, 25], and computational methods [22, 72, 81], as well as data comics authoring tools [38, 40, 69, 78]. Based on the design principles and literature survey, we derive several challenges, as listed below.

C1–Size and Complexity. Creating comics from dynamic graph data demands identifying key narrative elements, such as pivotal nodes or events, for storytelling. This task ranges from emphasizing critical nodes to elaborating on connections for specific events. Viewing dynamic graphs through various lenses, like individual nodes or node communities, can reveal these elements. However, analyzing dynamic graphs across multiple perspectives and timeframes is challenging due to the size and complexity of general dynamic networks.

C2–Identifying Characters. Because data comics are a genre of comics, they follow common design patterns in comics [50]. An important pattern is the existence of *main* and *supporting characters* [6, 42, 65, 70, 86]. Main characters drive the story while supporting characters play key roles in the narrative. Identifying which nodes serve as main or supporting characters poses a challenge, given the multitude of nodes with varying and evolving characteristics.

C3–Changes Over Time. The single most important task in a dynamic graph visualization is to show changes in the graph over time [1], particularly for the main and supporting characters, or their relationships. The challenge lies in detecting the changes, presenting them, and annotating the story. Detecting temporal changes requires a deep understanding of the data, and is difficult to do manually, especially for large graphs. Visualizing such changes is also not a trivial problem. Finally, annotations are essential for explaining the context or conveying additional insights.

C4–The Language of Comics. Comics employ a distinctive narrative structure, visual language, and temporal sequencing to engage readers. For end-users unfamiliar with these conventions, effectively integrating them into dynamic graph comics can be daunting. Ensuring these elements are used effectively to convey complex information in an accessible and compelling manner requires a deep understanding of comic artistry and narrative technique.

Design Requirements. Below we present design requirements (R1–R5) for our proposed data comics authoring tool that can help users overcome these challenges (C1–C4):

R1 Allow users to choose **temporal granularities** for the data comics that reflect their desired levels of detail (C1).

R2 Enable **multi-perspective storytelling** (individual, community-based, metric-based, etc) (C1).

R3 Help users **find main characters and supporters** for initiating a narrative (C2).

R4 Support users to **detect and present temporal changes** of dynamic graphs with appropriate annotations (C3).

R5 Provide **comics authoring mechanisms** to assist users in following genre conventions (C4).

4 Graph Comic Generation Techniques

We introduce techniques for streamlining the generation of graph comics. We develop a hierarchical clustering algorithm that groups snapshots into different temporal granularities based on their similarity. This clustering produces a dendrogram, enabling users to automatically create comics by selecting the number of panels.

4.1 Visualizing Dynamic Graphs as Comics

A *graph snapshot* is a static graph representing the state of a dynamic graph at a specific point in time. We can consider a dynamic graph as a collection of graph snapshots: snapshots organized in a temporal sequence, where a snapshot is generated whenever there is a change in the dynamic graph, such as the addition or deletion of a node or link. Such a dynamic graph can be naïvely visualized as a data comic by rendering one *comic panel* per graph snapshot and then visualizing the specific snapshot inside its panel as a node-link diagram or adjacency matrix. We provide continuity between adjacent panels by freezing the position of nodes from one panel to the next and emphasizing changes—node and link additions and deletions—using visual highlighting, such as colors, callouts, and visual effects. The result is a *graph comic* [4]: a sequence of panels showing a graph changing over time.

Unfortunately, this naïve approach is not practical for graphs spanning long time periods or involving many changes because the resulting comic will have a prohibitively large number of panels. Furthermore, large graphs with many nodes and links will yield panels that are so cluttered that individual changes are hard to spot. Below we describe practical approaches to address these shortcomings:

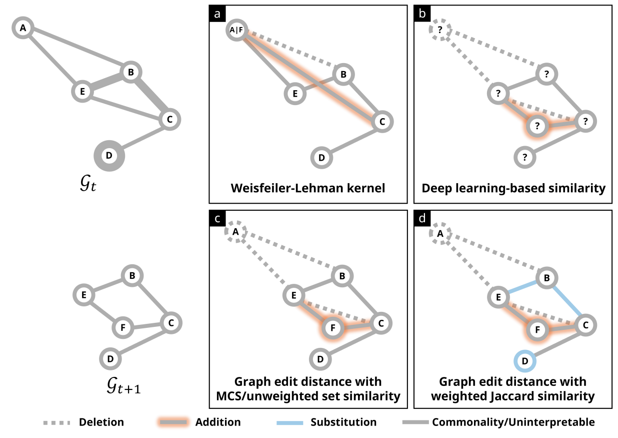
- **Long times:** We present a *causality-preserving temporal clustering* algorithm that combines multiple adjacent snapshots into clusters to support balancing the number of panels and the amount of changes in each panel based on the user's needs; and
- **Large graphs:** We propose graph filtering mechanisms based on the concept of *main* and *supporting characters* so that large graphs can be reduced to more manageable subgraphs.

4.2 Causality-Preserving Temporal Clustering

We present an algorithm that hierarchically groups graph snapshots in a dynamic graph based on the similarity between adjacent snapshots while preserving their causal (temporal) order. Instead of showing every graph snapshot, the graph comic can show *snapshot groups* consisting of the union of several temporally adjacent snapshots. The goal is to facilitate users precisely adapting the number of panels to show the resulting comic, from a single panel representing all changes (i.e., a snapshot group representing **all** snapshots) to a panel for every individual change in the graph. We consider four additional requirements for our algorithm: it must take into account the node and link labels, which are crucial for the graph data; it should preserve temporal continuity and consistency, meaning it should provide the same similarity for the same input; it should factor in multiple numerical attributes for both nodes and links; and it should not be computationally heavy.

4.2.1 Adjacency-based Hierarchical Clustering

We employ a variant of agglomerative clustering [24] where, instead of comparing the distance from each graph snapshot to every other snapshot, we only compare the distances between snapshots that are adjacent in time [85]. In other words, given a dynamic graph $\mathbb{G}$ consisting of a sequence of graph snapshots G_t for each time $t \in [0 \ldots T]$ for the



A
B
E
C
D
$\mathcal{G}_t$
$\mathcal{G}_{t+1}$
a
b
c
d
A|F
Weisfeiler-Lehman kernel
Deep learning-based similarity
Graph edit distance with MCS/unweighted set similarity
Graph edit distance with weighted Jaccard similarity
Deletion
Addition
Substitution
Commonality/Uninterpretable

Fig. 2: **Comparison of graph distance metrics for two graphs.** $\mathcal{G}_t$ and $\mathcal{G}_{t+1}$ with different labels and attributes with (a) Weisfeiler-Lehman kernel-based distance, (b) deep learning-based similarity (or distance), (c) graph edit distance with MCS or unweighted set similarity, and (d) graph edit distance with weighted Jaccard similarity (our choice). The thickness of the line is proportional to the node or link attributes.

time period T, only graph snapshots at $t-1$ and $t+1$ are compared (i.e., snapshots just before or after the current snapshot). This means that the temporal order is preserved when graph snapshots are aggregated into snapshot groups, thus maintaining causality in the resulting comic.

Conceptually, a *snapshot group* is regarded the same as a snapshot: it has a time span instead of a point in time representing the first and last time stamps for its constituent graph snapshots, and its graph is the union of those constituent graphs. Importantly, the distance metric is defined similarly for both a snapshot and a snapshot group.

To build a cluster hierarchy of a sequence of graph snapshots (and snapshot groups), we must provide a distance metric $D(G_1, G_2)$ that accepts two adjacent graph snapshots (or groups) G_1 and G_2. We agglomerate the dynamic graph sequence by progressively selecting the two adjacent snapshots with the smallest distance and replacing them with a snapshot group combining them. This means that the number of snapshots in the sequence will monotonically decrease until a single snapshot group representing all of the original graph snapshots remains.

4.2.2 Graph Edit Distance with Weighted Jaccard Similarity

Several graph similarity measures exist in the literature. Basic graph topology measures for unweighted, unlabeled, and undirected graphs are not good metrics of similarity, as real-world dynamic graphs often have both node and link attributes that factor into similarity. Next, we review existing work on graph distance measure metrics and propose our new distance measure metric used in this work.

We find four types of metrics: graph edit distance (GED), set similarity, kernel-based measures, and deep learning-based measures. GED calculates a weighted summation of predefined graph edit operators, such as insertion, deletion, and substitution of a node or link. Although simple and effective, GED requires human effort to determine each edit operator and the corresponding cost. Furthermore, existing GED methods define substitution as a link or node replacement, which is not suitable for real-world graphs [26]. For example, as shown in Fig. 2 (c), it cannot measure the attribute change of node D.

Set similarity can be used as a graph similarity metric based on the number of common nodes (or links) in two graphs [41, 73]. The main advantage of this method is that it has linear computational complexity and is suitable for real-world applications because of its label awareness. However, similar to GEDs, it cannot measure attribute changes.

Kernel-based methods using graph topology are another line of research for graph similarity computation. Examples include path-based kernels [27, 39], subtree pattern-based graph kernels [48, 58], and Weisfeiler-Lehman optimal assignment kernels [43]. While effective in measuring the overall similarity over the entire graphs, they often omit node attributes (R2) and cannot be used for varying aggregation levels (R2). In addition, as shown in Figure 2 (a), it may lose the label information (e.g., treating link (A, E) and (E, F) as the same one).

Deep learning-based methods [47, 82] have been developed to overcome the heavy computational costs in measuring graph similarity. However, the explainability of deep learning methods remains an open problem, so we do not use them in this work, as our graph comics need multi-perspective storytelling (R2) with solid reasoning (R3–R4).

After reviewing existing metrics and their characteristics, we find that traditional methods, including GED and set similarity, are the most appropriate metrics for our purpose. However, they cannot be directly used for our work, as they are not able to measure attribute changes. As such, we develop an enhanced method with GED and weighted Jaccard similarity. Our method calculates vector-form Jaccard similarity [62] for each common node or link. For example, as shown in Fig. 2, we find common elements with labels (e.g., D or (B, E)) and then calculate vector-form Jaccard similarity for the attributes of each common element. As a result, we can obtain the degree of changes per link or node (e.g., change of attributes for node D), as in Fig. 2 (d).

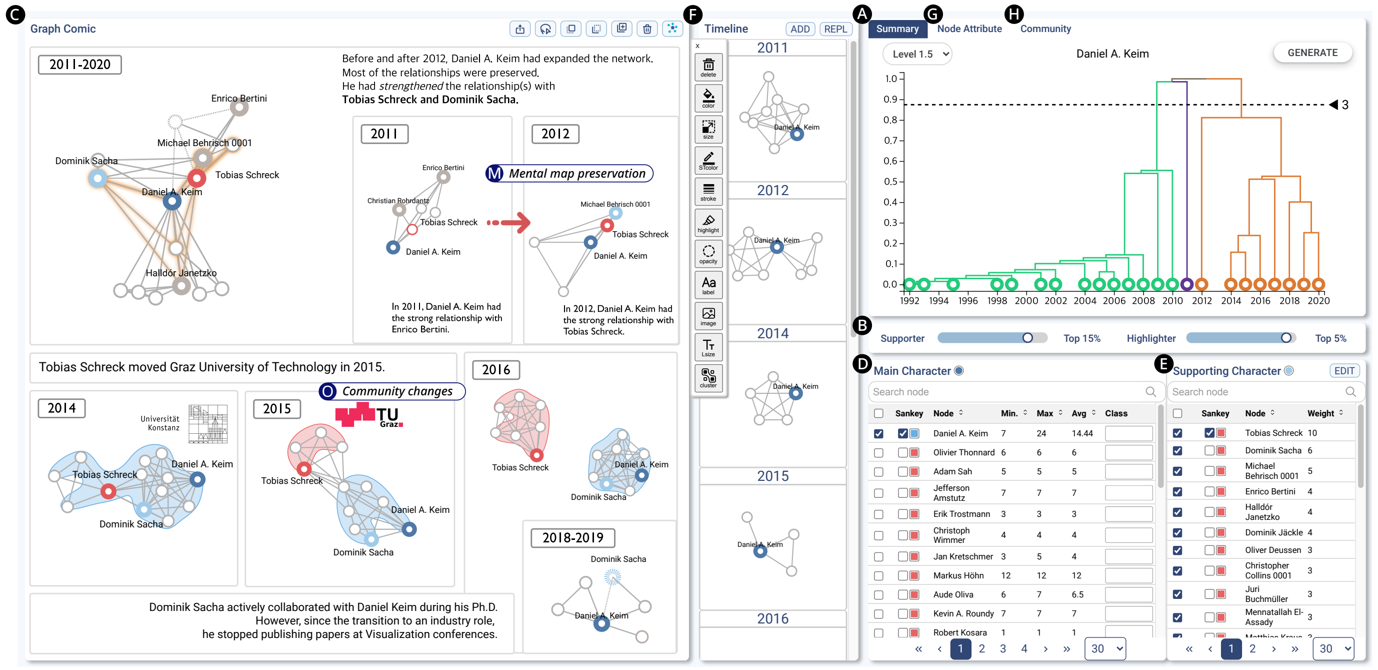
5 The DG Comics System

We design DG Comics to meet the challenges and satisfy the requirements from Sec. 3. Our approach outlined in Sec. 4 efficiently manages complexity (R1) for automatic graph comic generation. The algorithm output, visually represented as a dendrogram (Sec. 5.1), allows users to interactively choose an aggregation level for story fragments, which can then be depicted as comic panels (Sec. 5.2). Each comic panel represents a story fragment, and the node-link diagram in the panel visualizes changes during that interval (R4). Users can build their own stories by inspecting potential main and supporting characters (R3) with graph evaluation metrics (R2), such as node centrality, node degrees, and adjacency (Sec. 5.4), as well as individual graphs at different time points (Sec. 5.3). DG Comics facilitates the observation of communities (Sec. 5.5) where characters are involved (R2). Finally, it enables editing based on the language of comics (Sec. 5.2), including fonts, motion lines, captions, and layouts (R5). DG Comics employs Next.js for the frontend, FastAPI for the backend, and the D3 library [14] for visualizations, including computation of node-link diagram layouts. The source code is available at github.com/joohe-e/DGComics.git.

5.1 Summary View

Comics are made of *strips* of one or more *panels* [50, 64], each depicting a portion of the narrative and the sequence typically showing temporal progression. In a graph comic, each panel contains a snapshot of the dynamic graph sequence (either as an individual or a group) and a caption that provides information about the snapshot. The *Summary View* (Fig. 3A) facilitates the automatic generation of a series of panels via a dendrogram that visualizes the hierarchical clustering result (Sec. 4). Each leaf node of the dendrogram refers to a graph snapshot at a specific time point. The X-axis indicates the time from the beginning to the end of the data, and the Y-axis notes the normalized similarity between two adjacent snapshots from 0 to 1. The more similar the groups or individual snapshots are, the lower their position is connected, as the distance is shorter. A horizontal dashed line, called a **depth slider**, partitions the dendrogram into multiple clusters, visually encoded with different colors. The number beside the line indicates the number of clusters formed at that depth level.

Users can select the number of panels to include in the comic strip by moving the depth slider (R1). As users adjust the slider, the clusters are colored differently. They can generate the comic strip by clicking the GENERATE button. Each branch selected by the depth slider returns the characters that change the most over the timespan corresponding to the cluster (R3). For example, if a user clicks GENERATE after positioning the depth slider as shown in Fig. 3A, three panels are created in the comic view, each representing the assigned clusters (green, purple, and orange). DG Comics offers two options for constructing subgraphs with the ego being the resultant main character(s): a 1.0-level ego network, which includes the ego and its 1-degree alters, and a 1.5-level ego network which also includes the ties between the ego's alters. Users choose an option based on what relationship they want to present (R2).



Before and after 2012, Daniel A. Keim had expanded the network. Most of the relationships were preserved. He had *strengthened* the relationship(s) with **Tobias Schreck and Dominik Sacha.**

In 2011, Daniel A. Keim had the strong relationship with Enrico Bertini.

In 2012, Daniel A. Keim had the strong relationship with Tobias Schreck.

Tobias Schreck moved Graz University of Technology in 2015.

Dominik Sacha actively collaborated with Daniel Keim during his Ph.D. However, since the transition to an industry role, he stopped publishing papers at Visualization conferences.

☐	Sankey	Node	Min.	Max	Avg	Class
✓	✓ 🟦	Daniel A. Keim	7	24	14.44	
☐	☐ 🟥	Olivier Thonnard	6	6	6	
☐	☐ 🟥	Adam Sah	5	5	5	
☐	☐ 🟥	Jefferson Amstutz	7	7	7	
☐	☐ 🟥	Erik Trostmann	3	3	3	
☐	☐ 🟥	Christoph Wimmer	4	4	4	
☐	☐ 🟥	Jan Kretschmer	3	5	4	
☐	☐ 🟥	Markus Höhn	12	12	12	
☐	☐ 🟥	Aude Oliva	6	7	6.5	
☐	☐ 🟥	Kevin A. Roundy	7	7	7	
☐	☐ 🟥	Robert Kosara	1	1	1	

« ‹ 1 2 3 4 › » 30

☐	Sankey	Node	Weight
✓	✓ 🟥	Tobias Schreck	10
✓	☐ 🟥	Dominik Sacha	6
✓	☐ 🟥	Michael Behrisch 0001	5
✓	☐ 🟥	Enrico Bertini	4
✓	☐ 🟥	Halldór Janetzko	4
✓	☐ 🟥	Dominik Jäckle	4
✓	☐ 🟥	Oliver Deussen	3
✓	☐ 🟥	Christopher Collins 0001	3
✓	☐ 🟥	Juri Buchmüller	3
✓	☐ 🟥	Mennatallah El-Assady	3

« ‹ 1 2 › » 30

Fig. 3: **DG Comics overview.** DG Comics offers (A) a *Summary View,* (B) sliders for filtering and highlighting nodes, (C) a *Graph Comic View*, (D) *Main Character* and (E) *Supporting Character* tables, and (F) a *Timeline View*. Users can switch to (G) the *Node Attribute Table* or (H) *Community View* using the tab. It supports (M) *mental map preservation* by fixing nodes across displays, and (O) *community changes* using bubble sets.

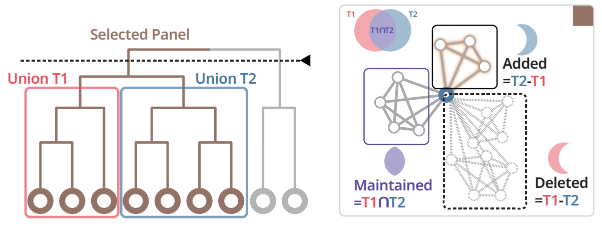


Fig. 4: **Data abstraction and presentation.** Each cluster below the depth slider (left) represents a panel (right) through a set operation of two subcluster snapshots. The aggregate graph snapshot is computed by the union of graphs for timespans T1 and T2.

5.2 Graph Comic View

The *Graph Comic View* allows users to edit graph comic templates while managing large dynamic graphs efficiently. To reduce visual clutter and enhance readability, we provide an option to keep only the top N percent of alters based on link weights visible. Users can control this level of visibility using **supporter slider**, and further highlight specific alters with the **highlighter slider**. Both sliders can be adjusted simultaneously before users begin editing (Fig. 3B).

5.2.1 Visual Design

We create a node-link diagram for each panel as the union of the panel's children (or the snapshot itself for leaf nodes). Each panel visualizes the changes between two consecutive children snapshot groups, highlighting added nodes and links with a neon effect and deletions with dashed strokes (R4). We abstract the data for each change using set operations: subtracting the former timespan (T1) from the latter (T2) yields the added subgraph, while the reverse reveals the deleted subgraph (Fig. 4). Supporting characters are defined as nodes with greater weight in their relationship with the main character. We emphasize main and supporting characters by thickening the stroke of the nodes, applying diverging colors, and including labels; their colors can be changed using a color picker.

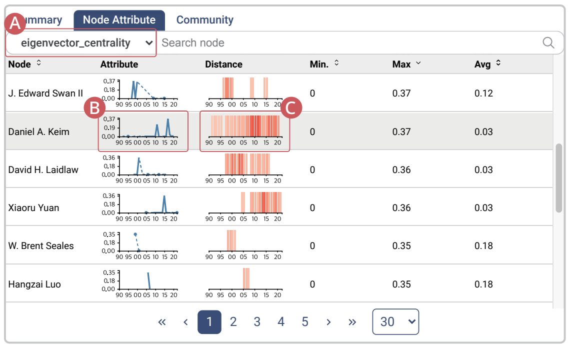
Panel Layout. Since each panel contains a narrative, its size and placement are crucial for conveying the message. Specifically, viewers can infer event duration from the width of the panel [64] and determine the chronological order based on the panel's position in the strip, where time proceeds from left to right and top to bottom [50, 64]. DG Comics is designed to generate an appropriate panel layout. The layout algorithm first calculates the number of rows, or *tiers*, by taking the square root of the total panel number specified by the user, ensuring each panel maintains a minimum height. For example, with 16 panels, DG Comics assigns 4 tiers. It then allocates panels to each tier to balance the sum of their time intervals. The comic is partitioned using this structure, with gutters between panels for clarity. This approach normalizes time points across tiers, assuming that panels with more time points convey more information, and encodes event duration into panel size [4].

Captions. DG Comics provides a caption template for each panel in the main character's point of view to assist storytelling. Each caption consists of three clauses about a summary, major change, and the most influential relationship. The summary describes expansions, contractions, and constancy based on the difference between the graphs before and after. It includes a prepositional phrase in front to indicate the timespan and reference time for a change. Then we compare the total number of nodes added, deleted, and preserved, and state one case that is the maximum. Lastly, we select the node(s) that have the highest link weights and mark whether the main character had obtained, lost, strengthened, or weakened its relationship with those nodes. A set of clauses is created per main character, but each clause can be combined into one with multiple subjects if different main characters share the same content. When the panel portrays a particular time point, it simply reports the strongest relationship.

5.2.2 Interaction Design

DG Comics offers diverse editing interactions to enhance flexibility in content creation (R5). Users can manage the general format of the canvas and graph drawing using the **toolbar** (Fig. 3C, top-right) in the *Graph Comic View* and adjust the details, such as text and graph styles within the canvas using the **toolbox** that appears on the right side of *Graph Comic View* when users interact with the element.

Canvas Editing. We define an editable component of the panel as a *canvas*. DG Comics includes three canvas types: graph, text, and image. An auto-generated panel contains one graph and two text canvases—a caption and temporal information. Users can move, resize, and rotate a canvas by interacting with the four sides and corners of the canvas, respectively. The canvas editing can be done using the toolbar on the upper right corner of the *Graph Comic View* (Fig. 3C). Clicking the ⧉ button reveals a list of canvas types that can be added to the workspace.



Node	Attribute	Distance	Min.	Max	Avg
J. Edward Swan II			0	0.37	0.12
Daniel A. Keim			0	0.37	0.03
David H. Laidlaw			0	0.36	0.03
Xiaoru Yuan			0	0.36	0.03
W. Brent Seales			0	0.35	0.18
Hangzai Luo			0	0.35	0.18

Fig. 5: **Node Attribute Table.** This table shows a list of all the nodes and their values over time.

One distinct option is Background, which inserts a background image into the graph canvas. To add a graph canvas, users select the timespan on the *Summary View* by brushing before pressing the Graph button. This will generate the change summary of an extracted main character during the chosen timespan as explained in Sec. 4.1. The Text option enables users to create an empty text box, while the Image option offers preset icons such as speech balloons and arrows and supports importing external images via the button. Users can simply delete, and move back and forth the focused canvas by clicking , , and buttons.

Graph Drawing. DG Comics provides multiple modes for generating the graph layout before manually repositioning. When pressing the button on the toolbar, buttons for three modes appear. The default mode is a Basic force-directed graph that uses the concepts of repulsion and attraction to place the relevant nodes closer and irrelevant nodes further. Derived from this graph structure, the Filled mode reduces empty space to a minimum in the way of multiplying scale to the relative position. This mode can be useful to increase the readability of nodes by preventing overlapping. The Fixed mode supports mental map preservation by fixing node positions across the panels (Fig. 3M). To achieve this, we compute a force-directed layout of the union of all nodes and links in the dynamic graph over its full timespan. This is useful for tracking changes in individual panels and facilitates the easy detection of nodes and links being added or deleted.

Text Editing. When the text canvas gets focused, it triggers a movable toolbox to appear on the upper right side of the *Graph Comic View*. Users can drag the text content for selection and change it to be bold, italic, or underlined. The typeface and size are also adjustable. The background and border of the text canvas can be on and off. We integrate an LLM (`Mixtral 8x7B` [36]); clicking the button will use the LLM to improve the caption automatically.

Graph Element Editing. In the default mode, users can zoom in on the graph canvas by scrolling and pan by dragging the background. Each node is movable, and its style, along with the link, is editable. DG Comics offers three methods for selecting multiple nodes: users can individually click on the nodes; employ the lasso tool from the graph comic view toolbar for easier group selection; or use a supporting character table by clicking on a character row and then the EDIT button (Fig. 3E). Once the selection is made, group movement of the nodes is available, and a toolbox appears, allowing customization of styles. The node-level toolbox provides functions for changing color, size, and stroke properties, adjusting opacity, applying highlights, adding labels, inserting images, deleting nodes, and clustering nodes. Links can be edited by a direct click, and the link-level toolbox offers similar functions, except for labeling, grouping, and image insertion. It also features the ability to change markers, particularly useful for directed graphs. Users can change the color and opacity of a cluster created either by manual grouping or by selecting from the *Community View* (Sec. 5.5). Once all the editing is done, users can export the final product as PNG, JPG, or PDF via the button.

5.3 Timeline View

We can express temporal changes not only with the symbolic representation but also through the separation of panels. The *Timeline View* (Fig. 3F) lets users choose a transition style by listing static graphs of different time points. Users can scrutinize graphs included within the chosen time span and select one to add to the canvas (R4, R5).

The *Timeline View* provides two functions for generating a series of graphs: adding and replacing the selected panel. The ADD option enables switching focus from overview to detail by attaching the graphs of certain points in the bottom right corner. We use the same layout computation for the ADD option to embed multiple panels in a limited space while not covering the root panel completely. The REPL option replaces the selected panel and arranges new panels in temporal order.

5.4 Tables

We provide three tables in DG Comics to select node attributes and characters to display in the graph comic. The common features of the tables are sorting by values and searching by names.

The *Node Attribute Table* (Fig. 5) facilitates node exploration and assists in manual main character selection by providing comprehensive attribute values and chart thumbnails (R3). Users can switch to this table by clicking the second tab above the *Summary View* (Fig. 3G). The line chart thumbnail tracks the value over time; lines are dashed if data is missing (Fig. 5B). To prevent possible bias from inconsistent scales, we also mark minimum, maximum, and average values. Users can examine different attributes of a node—e.g., the total number of links, PageRank, and centrality—by selecting the option in the top left corner of the view (Fig. 5A). The heatmap presents the degree of dissimilarity between the consecutive time points, meaning that the color gets darker as the distance computed with our modified Jaccard index increases (Fig. 5C). Using this thumbnail, users can capture moments with significant change. When selecting a main character on the attribute table, the dendrogram is re-created to reflect the subgraph-level change (R2). This means that each time node of the dendrogram represents the subgraph grounded in the selected node (Fig. 3A).

The character tables (Fig. 3D, E) manage the nodes depicted in the graph comic. The list of nodes is updated whenever users click the graph canvas to focus. Users can either add or delete nodes by selecting or deselecting the leftmost checkbox. While the *Main Character Table* (Fig. 3D) shows all the nodes as candidates for the main character, the *Supporting Character Table* (Fig. 3E) lists the neighbor nodes that are linked with the currently selected main character (R3). The minimum, maximum, and average values in the *Main Character Table* reflect the attribute selected in the *Node Attribute Table*. The total weight of links to a node in the selected time span is an indication of the importance of the node (Fig. 3E). The *Class* column in the *Main Character Table* enables direct manipulation of graph element styles. By entering the tag name as input and clicking on the *Class* header, users can open the CSS editing view to redefine the properties of the corresponding nodes (Fig. 3D). The EDIT button in the *Supporting Character Table* is explained in Section 5.2.2.

5.5 Community View

Identifying the communities in which nodes are involved (i.e., community membership) and tracking their evolution is crucial for a deeper understanding of their relationships (R2). For instance, if two nodes from different communities merge into the same community, we can assume that their relationships become stronger; if they split, the relationship becomes weaker. A Sankey diagram is a suitable approach to illustrate such flow by representing communities as nodes and transitions between them as links. To differentiate their nodes and links from graph elements, we call them *community nodes* and *paths*. However, its readability significantly decreases with the numerous overlapping paths and varied sizes of community nodes. Given that the narrative of the graph comic is character-driven, we decided to highlight only the communities of the chosen characters in this work. Moreover, we re-designed the Sankey diagram to enable fair inspection and comparison of community changes associated with the characters. Fig. 6 shows our *Community View*, re-designed in this work.

In a traditional Sankey diagram, the size of a community node is proportional to the size of the community, making it difficult to observe transitions for characters in smaller communities. To overcome this

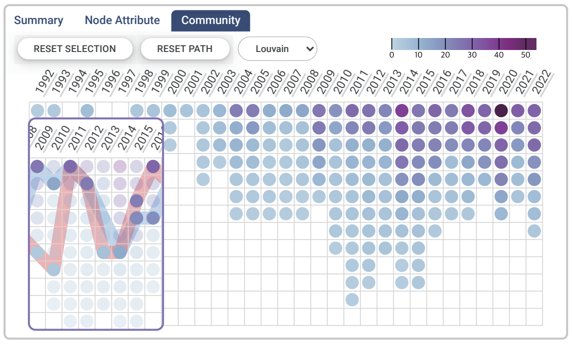


Fig. 6: **Community View** changes when communities are selected, as shown in the left inset, where red and blue colors mean Keim and Schreck, respectively, whose communities diverged since 2014.

obstacle, we encode the community size into a sequential color scheme while normalizing the community node size. Users can choose a cluster color in a color palette by selecting the second checkbox on the 'Sankey' column (Fig. 3 D, E). We set the time as the x-axis and place community nodes from top to bottom based on their community size. To connect the community nodes for a selected character, we adopt a circular shape with padding inside the grid cell so that the path can enclose it smoothly. As a result, the *Community View* displays the colored path of community evolution for the selected character while blurring nodes outside of this path as Fig. 6 shows.

The *Community View* enables users to detect six types of events in community evolution (Fig. 7): growth, contraction, merging, splitting, birth, and death [56]. Since DG Comics supports character-based exploration for generating narratives, the community is primarily identified by the character. Therefore, the start and end of the path indicate the birth and death of the character's community within the dataset's time span. Users can determine growth and contraction through color changes and the vertical position of the circle across different years. Furthermore, by selecting multiple characters, users can observe how distinct communities evolve to merge and split at specific time points throughout the time span. Notably, the visual patterns depicted by multiple paths can illustrate how the communities evolved through various events (e.g., splitting, merging, homogeneity, heterogeneity) over different time periods as Fig. 7 shows. After exploring community evolution, users can add a cluster to the canvas by clicking the community node (Fig. 3O). DG Comics uses Bubble Sets [21] to highlight the group of characters involved. The RESET PATH button (upper left) clears all highlights from the *Community View*, while the RESET SELECTION button removes all the clusters from the graph comic. The default setting displays the communities detected by the Louvain algorithm, but another option is available in the drop-down list, such as affiliations of authors in the case of Vispubdata.

6 Example

We demonstrate the use of DG Comics in scientometric analysis with data from the visualization community. We use Vispubdata [35], a dataset containing information on 3,620 papers from IEEE Visualization conferences (InfoVis, SciVis, VAST, and VIS) from 1990 to 2022, including conference, year, title, paper type, authors, affiliation, keywords, and number of citations. This dataset is frequently used for cross-checking in dynamic network research and is highly relevant to the visualization community. To create graph comics with compelling storylines, we constructed author networks, where nodes represent authors and links represent collaborations based on co-authored papers.

6.1 Analysis

We start with the default dendrogram, which computes the dissimilarity-based hierarchy of the entire dynamic graph, adjusting the depth slider to segment the time into nine divisions. This process generates nine panels, offering a visual narrative of the data, which in this case, is the collaboration network in VIS papers. We select 1.5-level ego networks to observe connections between the neighbor nodes as well.

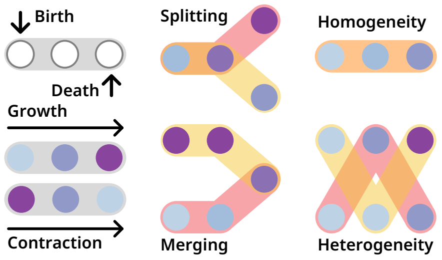


Fig. 7: **Community evolution.** Six archetypes of community evolution.

The auto-generated comic reveals Daniel A. Keim as a central figure in two distinct periods, 2006–2011 and 2012–2016, with the network expanding in the earlier period and contracting in the latter. We can now generate a graph comic from Keim's perspective for closer examination.

By opening the *Node Attribute Table* and searching for "Keim," we detect the high dissimilarity (Fig. 5C) and peak in eigenvector centrality (Fig. 5B) between 2011 and 2012, which suggests transformations in Keim's collaborative patterns. With this insight, we create a new dendrogram for Keim and regenerate the graph comic, which unveils two main clusters before and after 2011, indicating major events in the network's evolution (Fig. 3A). We move the sliders to filter out impactful relationships involving the top 15 percent of total collaborations and highlight the top 5 percent (Fig. 3B). To examine changes closely, we add the graph canvas by brushing 2011 to 2020 from the *Summary View* and insert the individual graphs of 2011 and 2012 from the *Timeline View* (Fig. 3F). The tool's features allow us to preserve the mental map (Fig. 3M), helping us to capture the establishment of new relationships and community shifts. Notably, we observe a strengthening of relationship with Tobias Schreck from 2011 to 2020, prompting us to further compare the dynamics between Keim and Schreck.

In the *Community View*, we color-code Keim's (blue) and Schreck's (red) communities and draw paths to visualize their changes over time (Fig. 6). Despite multiple collaborations, their communities have diverged since 2014 (Fig. 3O), reflecting the impact of Schreck moving to Graz University of Technology in 2015. This divergence, despite ongoing collaboration, illustrates nuanced dynamics common in academic networks. Further exploration reveals Dominik Sacha as a significant yet eventually diverging connection, mirroring the evolution of community affiliations and the impact of career milestones such as Ph.D. completion and industry employment. By juxtaposing Sacha's trajectory with Schreck's, we enrich the narrative, offering comparative insights into how individual careers and collaborations shape the broader academic community. This scenario demonstrates the tool's capacity to dissect, visualize, and narrate complex relational data, making it an invaluable resource for understanding dynamic networks.

7 Evaluation

To explore the usefulness and effectiveness of DG Comics, we conducted a comprehensive user study. This study includes usability tests and interviews to gather valuable insights from participants and in-depth feedback from experts to validate its versatile application.

7.1 User Study

We conducted a controlled user study to evaluate DG Comics. The study consisted of a pre-study questionnaire, a tutorial session, graph comics creation, and a post-questionnaire with interviews. We present both quantitative ratings and qualitative insights from the interviews. We used the COW trade dataset [9, 10] of international trades from 1870 to 2014, covering 207 unique nations. Using this data, we built a directed graph of international trade where each node and link represents nationality and corresponding trade (in U.S. dollars), respectively.

7.1.1 Participants

We recruited 13 participants from a university (4 females) aged from 23 to 29 ($M = 25.4, SD = 2.2$). The number of participants is consistent with prior work [17, 34, 54], aligning with recommendations for usability testing. They were graduate ($n = 8$) and undergraduate ($n = 5$) students from electrical engineering ($n = 4$), computer science ($n = 5$),

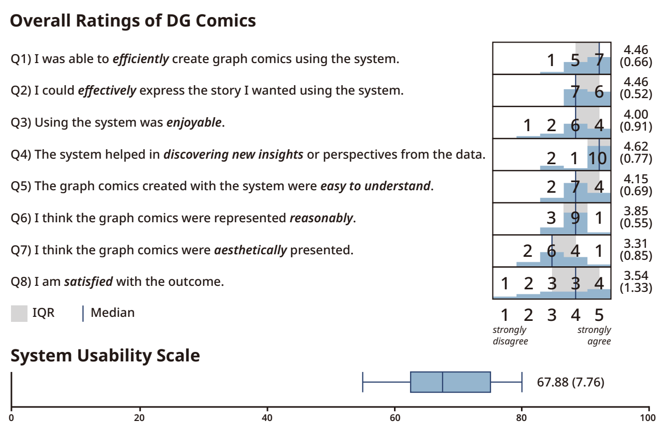


Fig. 8: **DG Comics ratings.** Overall ratings (top) and SUS score (bottom). Values are means and standard deviations (in parentheses).

and artificial intelligence ($n = 4$). All participants had experience in analyzing data using Python and visualizing it with Python, Excel, and PowerPoint. Most of them had experience dealing with graph data (e.g., traffic, sensor network, etc.), except for three participants who were aware of graph data but lacked experience in processing it. Participants were compensated $22 (USD) for 2 hours of study. The study was reviewed and deemed exempt by our Institutional Review Board.

7.1.2 Procedure

The study began with a pre-study questionnaire that included: (1) demographic information, such as age, gender, and education level; (2) participant experience with data analysis, data visualization, and graph data; and (3) participant familiarity with graph comics, the trade dataset, and events of world history relevant to the study. We then explained what a data comic is and how it can be applied to graph data. We conducted a 20-minute tutorial on how to use DG Comics and interpret the visual representations. Then participants were given 5 to 10 minutes to explore the system on their own and ask questions about its functions. Note that we used Vispubdata [35] for the tutorial session to prevent learning effects on the dataset used in the experiment.

After participants felt comfortable with the tool, we introduced them to an international trade dataset spanning from 1930 to 1960. They were tasked with using DG Comics to craft an engaging graph comic, highlighting the evolving relationships between nations. To aid in their task, we provided access to a Wikipedia page titled *Timelines of Modern History*, which includes sections such as *Timeline of the 20th Century* and *List of Wars*. The purpose was to accommodate varying levels of familiarity with world history among participants. While participants could use search engines to gather information and craft their storylines, we prohibited the use of any comprehensive historical resources, such as videos, that might present a finished interpretation of events.

Assistance was offered solely at the request of participants, ensuring that their interaction with the system remained independent. Participants used Google Chrome in full-screen on two 32-inch monitors with a resolution of 2560×1600 pixels—one for information search and the other for the actual comic creation. Participants were allotted 1 hour and 10 minutes for this task, during which their interactions with the system were screen-recorded for further analysis.

Following the comic creation phase, we asked participants to fill out a post-study questionnaire. This questionnaire gauged their opinions on the usability, overall experience, and the helpfulness of different views within DG Comics, as well as their satisfaction with their final product, using a 5-point Likert scale. We concluded each session with a semi-structured interview to explore the reasoning behind questionnaire responses, the narratives constructed, and comprehensive feedback on the system's functionality and user experience.

7.1.3 Quantitative Results

We adopted a 5-point Likert scale to assess the overall experience of the system (Q1-Q4) and the output that participants generated (Q5-Q8). We used the System Usability Scale (SUS) designed by Brook et al. [15] for global evaluation of systems usability. Interpreting the Likert scales as ordinal, we report the quantitative results in a discrete visualization style [67], the histogram [32, 84] with medians (as blue lines) and quartiles (in gray areas). We also provide means and standard deviations considering intervalist views [19].

As Fig. 8 shows, participants felt positive about their experience using the system (Q1-Q3). In particular, they highly valued discovering new insights and perspectives from the system (Q4). They all agreed that the comics they made were easy to understand (Q5-Q6). However, aesthetics and satisfaction with the output varied among the participants, likely due to the discrepancy between their editing skills and expectations toward their outcomes (Q7-Q8). The SUS score is acceptable ($M = 67.88, SD = 7.76$); compare this to the average of 68.2 from 273 prior studies [8].

7.1.4 Qualitative Results

We analyzed participants' responses and video recordings associated with the scores. Here we present these qualitative results.

The summary view supports efficient story generation. Most participants reported that the overview with the dendrogram helped generate stories ($MD = 4, M = 4.15, SD = 0.99$). Even though most of them had little knowledge of the data, they were able to understand the summary of changes in relationships at a glance. They used the auto-generated comic as a starting point. The hierarchical clustering effectively grouped similar time points to depict relevant stories into one panel, allowing participants to "*choose how detailed or brief to express the story by adjusting the level* (P5)." Clusters gave a hint to users where to focus; for example, P9 after detecting groups before and after World War II, "*focused on the specific branches for storytelling the events*" while P12 "*scrutinized the clusters as a chunk of changes*."

The community view effectively aids in inferring relationships between nodes and events. Participants used the community view to check the relationship between nodes ($n = 7$). Some participants displayed a node's community to see other nodes in the same cluster and used this information to structure the graph (P3, P7, P8, P11, P12). P12 excluded some countries such as the U.S. from the graph since "*their consistency in the relationship with the United Kingdom suggests no special event happening*." They instead captured insights by inspecting the community visualization. For example, the comparison between the community evolution paths of different countries provides an implication for a significant event when there is a split or a merge. P6 searched the interaction between Poland and Germany since "*their paths overlapped during the 1930s*."

Participants wanted to use the system with their own data. All participants indicated that the system helped them effectively express the story ($MD = 4, M = 4.46, SD = 0.52$). Some commented on challenges in visualizing dynamic graphs, especially for big and complex data (P1, P5). However, using DG Comics, they were able to convey changes in the relationship between nodes over time regardless of familiarity with the data ($n = 7$). P7 noted that it became much easier to narrate the story of network evolution because "*the system helps with the hardest part, [which is] selecting the main character.*" P10 said that although he did "*not have an aesthetic sense, [he liked that he could] make a satisfactory product within an hour*." Some participants brought up the point that the convenience of the system does not only rely on visualization but also on interpretation of the data. "*If the series of time points are clustered, users can think about the reason and assume the circumstance* (P5)." P1, who conducts research on tracking individuals using ultra-wideband data, said, "*it would be very helpful to [be able to] detect when interference between two groups occurs*."

7.2 Expert Feedback

We conducted online interviews with six experts across various domains, who were invited via emails with consideration of domains and expertise. Each expert has a Ph.D. degree and 8 to 19 years of work experience in their respective fields. Their areas of expertise include Communication (E1: 19 years, E2: 8 years), History (E3: 12 years), Design (E4: 12 years), Management Information System (E5: 8 years), and Economics and Finance (E6: 11 years). Before interviews, we sent them the information on the interviews (e.g., goal, duration, expected outcomes), a demo video, and a website link to the deployed system. The interviews began with a short system demonstration, followed by a Q&A session for any clarifications. Once interviewees were familiar with the system, we proceeded to discuss its various aspects.

All appreciated the usefulness of the tool in telling a story using an approachable format (i.e., comic). They reported various strengths of DG Comics, including automated story extraction, rich functionality in editing for story presentation, and novelty of the approach in DG Comics. E1 lauded "*the system is capable of extracting outstanding stories with the corresponding timespan, streamlining the research process for data journalists.*" E4 and E5 also noted that laypeople, such as marketers or end-user creators, can use the tool to create and present a story with data that would otherwise have been inaccessible to them. E1 and E2 raised a similar point, that DG Comics enriches users' insights by providing the change in dynamics between characters. E3 even expressed his willingness to incorporate the system into his curriculum of political history. "*Given the possible impact of politics on economics, I believe students can gain insights into economic relations, such as international trade, by comparing them with political knowledge through a comic."* E5 and E6 underlined the contribution of autogenerated output to efficient decision-making. Specifically, E5 commented that autogenerated output from dynamic graphs will help users in management who need to analyze dynamically changing relationships among companies—"*an intuitive understanding of supply flows and the impacts of subcontractor changes can help users make strategic decisions in supply chain management."* E4 noted that many people need to create and present stories from data but are often frustrated by challenges brought by new and complex data. In this situation, he expected the tool would be very helpful. He highlighted the novelty of the approach that DG Comics provides, stating, "*... I give 10 out of 10 for novelty because I have not seen this type of system and support from a complex data storytelling perspective.*"

All experts estimated that the target user spectrum is broad, encompassing a diverse array of individuals. With the automatic generation of comic templates and robust editing functions for crafting user-driven stories, experts believe DG Comics meets the needs of a wide range of users, from novices to professionals. They also commented that usage scenarios with DG Comics may vary depending on the user's level of expertise and goals, even among the same expert group. For instance, E1 predicted that "*novice users would primarily utilize the Graph Comic View, whereas experts would navigate between views to gain a deeper understanding of the data.*" Additionally, E4 mentioned that "*researchers and data analysts would review the summary of changes from a generated comic to select data before beginning their analysis, yet if they already know the key points, their focus would shift toward authoring to convey their message.*"

Some experts expressed concerns about the steep learning curve due to the comprehensive features implemented in the system. For example, E2 and E4 expressed a concern that this system seems to require some learning time before users fully enjoy the system functions. They suggested creating a lite version of the system with fewer features for beginners. On the other hand, some other experts, namely E2, E5, and E6, indicated that experts may need additional features that help users automatically process unstructured, complex graph datasets. E5, in particular, mentioned that there is a significant amount of dynamic graph data in companies, but only data scientists can pre-process the data for analysis and presentation purposes—"*If the tool included an interface for processing raw dynamic graph data, it could have a more substantial impact, given the number of employees in enterprises who need to analyze, monitor, and present stories with large graph datasets.*"

8 Limitations and Future Work

DG Comics provides a flexible and intuitive authoring environment for graph comics creation. However, the numerous functions and steps in the authoring process may cause newcomers to face a steep learning curve. To mitigate this, providing an interactive tutorial and a help menu that supports keyword search and mouse-over explanations can be a reasonable solution. We also consider providing a function for adjusting the level of interface complexity. Since our user study may not capture the full diversity of all potential users and was conducted in a limited amount of time, further investigation, such as a longitudinal study, would help ensure ecological validity in the complexity reduction of visualization and storytelling tools across different users.

The wide range of domain experts we interviewed suggests that DG Comics has potential for diverse applications. However, questions about data preprocessing remain. Expert feedback reveals that real-world network data, such as transactions between companies, communications within and between departments, and distribution channels, are often unstructured and unprocessed. To address this, automated preprocessing of raw data into a compatible format is the next crucial step for DG Comics to enhance its versatility. As a prototype, DG Comics does not provide automated methods for extracting interesting patterns from node and link attributes (e.g., anomaly detection for time series [66]). Future work may focus on developing automated methods to improve the utility and generalizability of storytelling tools. We also expect the tool can be extended to incorporate datasets from biology (DNA, proteins), software engineering (call graphs, code dependencies, developer activities), and road networks with varying traffic situations.

While the dendrogram effectively encodes each snapshot as a leaf node, it faces scalability limitations with large dynamic networks containing numerous snapshots. To enhance readability, we need to explore abstraction techniques [44, 52, 57] such as collapsing and expanding snapshots, despite the additional interaction they introduce. Currently, the prototype supports only a linear cut on the dendrogram. Though users can create additional canvases by brushing, enabling multiple cuts at different abstraction levels would offer greater flexibility in generating templates automatically. A future system supporting automated story generation with multiple cuts at different levels would improve both scalability and usability for real-world applications.

Besides, examining more sophisticated filtering techniques and community discovery methods would significantly enhance functionality. DG Comics offers sliders for intuitive filtering, allowing users to highlight important nodes based on link weights and reduce visual clutter. However, adopting more advanced filtering techniques should be inspected to prevent potential biases [60] from selecting subgraphs centered on the main character. The *Community View* displays communities detected by the Louvain algorithm or predefined communities such as affiliation. To enrich narratives, future work could incorporate advanced community discovery methods for temporal networks [61], enabling the tool to suggest more intriguing and comprehensive stories.

9 Conclusion

This paper presents DG Comics, a semi-automatic authoring tool for graph comics based on identifying notable changes in a dynamic graph. Our approach automatically extracts meaningful events using a causality-preserving hierarchical clustering method. With the algorithmic output, users can adjust granularities by manipulating the aggregation level and refine aesthetics by making editorial decisions down to minute details in comic strips through an intuitive user interface. Our user evaluation and expert feedback demonstrate the usefulness of the system, providing promising areas for future work.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.RS-2023-00218913, No. 2021R1A2C1004542), by a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number:HI22C0646), and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants (No. 2024-00360227-Leading Generative AI Human Resources Development, No. 2020-0-01336–Artificial Intelligence Graduate School Program, UNIST). Niklas Elmqvist was funded by Villum Investigator grant VL-54492 by Villum Fonden.

References

[1] J.-w. Ahn, C. Plaisant, and B. Shneiderman. A task taxonomy for network evolution analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):365–376, 2014. doi: 10.1109/TVCG.2013.238 2, 3

[2] D. Archambault, T. Munzner, and D. Auber. Grouseflocks: Steerable exploration of graph hierarchy space. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):900–913, 2008. doi: 10.1109/TVCG.2008.34 2

[3] B. Bach, N. Henry-Riche, T. Dwyer, T. Madhyastha, J.-D. Fekete, and T. Grabowski. Small MultiPiles: Piling time to explore temporal patterns in dynamic networks. *Computer Graphics Forum*, 34(3):31–40, 2015. doi: 10.1111/cgf.12615 2, 3

[4] B. Bach, N. Kerracher, K. W. Hall, S. Carpendale, J. Kennedy, and N. H. Riche. Telling stories about dynamic networks with Graph Comics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 3670–3682. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858387 1, 2, 3, 5

[5] B. Bach, E. Pietriga, and J.-D. Fekete. Visualizing dynamic networks with matrix cubes. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 877–886. ACM, New York, NY, USA, 2014. doi: 10.1145/2556288.2557010 2

[6] B. Bach, N. H. Riche, S. Carpendale, and H. Pfister. The emerging genre of data comics. *IEEE Computer Graphics and Applications*, 37(3):6–13, 2017. doi: 10.1109/MCG.2017.33 2, 3

[7] B. Bach, Z. Wang, M. Farinella, D. Murray-Rust, and N. H. Riche. Design patterns for data comics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 1–12. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173612 2, 3

[8] A. Bangor, P. Kortum, and J. Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009. 8

[9] K. Barbieri, O. M. Keshk, and B. M. Pollins. Trading data: Evaluating our assumptions and coding rules. *Conflict Management and Peace Science*, 26(5), 2009. 7

[10] K. Barbieri and O. M. G. Keshk. Correlates of war project trade dataset codebook, version 4.0, 2016. `https://correlatesofwar.org/`. 7

[11] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1):133–159, 2017. doi: 10.1111/cgf.12791 2, 3

[12] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete. Matrix reordering methods for table and network visualization. *Computer Graphics Forum*, 35(3):693–716, 2016. doi: 10.1111/cgf.12935 2

[13] A. Bezerianos, P. Dragicevic, J. Fekete, J. Bae, and B. Watson. GeneaQuilts: A system for exploring large genealogies. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1073–1081, 2010. doi: 10.1109/TVCG.2010.159 2

[14] M. Bostock, V. Ogievetsky, and J. Heer. D^3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. 4

[15] J. Brooke et al. Sus: A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996. 8

[16] M. Burch, M. Hlawatsch, and D. Weiskopf. Visualizing a sequence of a thousand graphs (or even more). *Computer Graphics Forum*, 36(3):261–271, 2017. doi: 10.1111/cgf.13185 2

[17] K. Caine. Local standards for sample size at chi. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 981–992. ACM, New York, NY, USA, 2016. 7

[18] E. Cakmak, U. Schlegel, D. Jäckle, D. Keim, and T. Schreck. Multiscale snapshots: Visual analysis of temporal summaries in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):517–527, 2020. doi: 10.1109/TVCG.2020.3030398 2, 3

[19] J. Carifio and R. Perla. Resolving the 50-year debate around using and misusing likert scales. *Medical education*, 42(12):1150–1152, 2008. 8

[20] Q. Chen, S. Cao, J. Wang, and N. Cao. How does automation shape the process of narrative visualization: A survey on tools. *IEEE Transactions on Visualization and Computer Graphics*, 30(8):4429–4448, 2024. doi: 10.1109/TVCG.2023.3261320 2

[21] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009. doi: 10.1109/TVCG.2009.122 7

[22] A. Dal Col, P. Valdivia, F. Petronetto, F. Dias, C. T. Silva, and L. G. Nonato. Wavelet-based visual analysis of dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(8):2456–2469, 2017. doi: 10.1109/TVCG.2017.2746080 2, 3

[23] C. Dunne and B. Shneiderman. Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 3247–3256. ACM, New York, NY, USA, 2013. doi: 10.1145/2470654.2466444 2

[24] N. Elmqvist and J. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2010. doi: 10.1109/TVCG.2009.84 2, 3

[25] V. Filipov, A. Arleo, and S. Miksch. Are we there yet? A roadmap of network visualization from surveys to task taxonomies. *Computer Graphics Forum*, 42(6):e14794, 2023. doi: 10.1111/cgf.14794 2, 3

[26] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis and Applications*, 13, 2010. 4

[27] T. Gartner, P. Flach, S. Wrobel, B. Scholkopf, and M. Warmuth. On graph kernels: Hardness results and efficient alternatives. In *Proceedings of Learning Theory and Kernel Machines*, vol. 2777, pp. 129–143, 2003. 4

[28] S. Ghani, N. H. Riche, and N. Elmqvist. Dynamic insets for context-aware graph navigation. *Computer Graphics Forum*, 30(3):861–870, 2011. doi: 10.1111/j.1467-8659.2011.01935.x 2

[29] S. Hadlak, H. Schumann, C. H. Cap, and T. Wollenberg. Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2267–2276, 2013. doi: 10.1109/TVCG.2013.198 2

[30] N. Henry and J.-D. Fekete. MatrixExplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):677–684, 2006. doi: 10.1109/TVCG.2006.160 2

[31] N. Henry, J.-D. Fekete, and M. J. McGuffin. NodeTrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, 2007. doi: 10.1109/TVCG.2007.70582 2

[32] J. Huang, A. Mishra, B. C. Kwon, and C. Bryan. Conceptexplainer: Interactive explanation for deep neural networks from a concept perspective. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):831–841, 2022. doi: 10.1109/TVCG.2022.3209384 8

[33] J. Hullman, S. M. Drucker, N. H. Riche, B. Lee, D. Fisher, and E. Adar. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2406–2415, 2013. doi: 10.1109/TVCG.2013.119 3

[34] W. Hwang and G. Salvendy. Number of people required for usability evaluation: the 10$\pm$2 rule. *Comm. of the ACM*, 53(5):130–133, 2010. 7

[35] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko. vispubdata.org: A Metadata Collection about IEEE Visualization (VIS) Publications. *IEEE Transactions on Visualization and Computer Graphics*, 23, 2017. doi: 10.1109/TVCG.2016.2615308 7, 8

[36] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024. doi: 10.48550/ARXIV.2401.04088 6

[37] B. Kale, M. Sun, and M. E. Papka. The state of the art in visualizing dynamic multivariate networks. *Computer Graphics Forum*, 42(3):471–490, 2023. doi: 10.1111/cgf.14856 3

[38] D. Kang, T. Ho, N. Marquardt, B. Mutlu, and A. Bianchi. ToonNote: Improving communication in computational notebooks using interactive data comics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 1–14. ACM, New York, NY, USA, 2021. doi: 10.1145/3411764.3445434 2, 3

[39] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the International Conference on Machine Learning*, p. 321–328, 2003. 4

[40] N. W. Kim, N. H. Riche, B. Bach, G. Xu, M. Brehmer, K. Hinckley, M. Pahud, H. Xia, M. J. McGuffin, and H. Pfister. DataToon: Drawing dynamic network comics with pen + touch interaction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 1–12. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300335 2, 3

[41] P. M. Kogge. Jaccard coefficients as a potential graph benchmark. In *Proceedings of IEEE International Parallel and Distributed Processing*

Symposium Workshops, pp. 921–928, 2016. 4

[42] R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. *IEEE Computer*, 46(5):44–50, 2013. doi: 10.1109/MC.2013.36 3

[43] N. M. Kriege, P.-L. Giscard, and R. C. Wilson. On valid optimal assignment kernels and applications to graph classification. In *Advances in Neural Information Processing Systems*, p. 1623–1631, 2016. 4

[44] B. Lee, C. S. Parr, C. Plaisant, B. B. Bederson, V. D. Veksler, W. D. Gray, and C. Kotfila. Treeplus: Interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414–1426, 2006. doi: 10.1109/TVCG.2006.106 9

[45] B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale. More than telling a story: A closer look at the process of transforming data into visually shared stories. *IEEE Computer Graphics and Applications*, 35(5):84–90, 2015. doi: 10.1109/MCG.2015.99 3

[46] W. Li, S. Schöttler, J. Scott-Brown, Y. Wang, S. Chen, H. Qu, and B. Bach. Networknarratives: Data tours for visual network exploration and analysis. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3581452 2

[47] G. Ma, N. K. Ahmed, T. L. Willke, and P. S. Yu. Deep graph similarity learning: a survey. *Data Mining and Knowledge Discovery*, 35(3):688–725, 2021. doi: 10.1007/S10618-020-00733-5 4

[48] P. Mahe and J.-P. Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75:3–35, 2009. 4

[49] T. May, M. Steiger, J. Davey, and J. Kohlhammer. Using signposts for navigation in large graphs. *Computer Graphics Forum*, 31(3pt2):985–994, 2012. doi: 10.1111/j.1467-8659.2012.03091.x 2

[50] S. McCloud. *Understanding Comics: The Invisible Art*. William Morrow Paperbacks, 1994. 2, 3, 4, 5

[51] F. McGee, M. Ghoniem, G. Melançon, B. Otjacques, and B. Pinaud. The state of the art in multilayer network visualization. *Computer Graphics Forum*, 38(6):125–149, 2019. doi: 10.1111/cgf.13610 2, 3

[52] K. Misue. Area-adaptive drawing of rooted trees. In *Proceedings of the IEEE Pacific Symposium on Visualization*, pp. 152–161, 2024. 9

[53] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal. graph2vec: Learning distributed representations of graphs. *CoRR*, abs/1707.05005, 2017. 2

[54] J. Nielsen. *Usability engineering*. Morgan Kaufmann, 1994. 7

[55] C. Nobre, M. Meyer, M. Streit, and A. Lex. The state of the art in visualizing multivariate networks. *Computer Graphics Forum*, 38(3):807–832, 2019. doi: 10.1111/cgf.13728 2

[56] G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007. doi: 10.1038/nature05670 7

[57] C. Plaisant, J. Grosjean, and B. B. Bederson. Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proceedings of the IEEE Symposium on Information Visualization*, pp. 57–64, 2002. doi: 10.1109/INFVIS.2002.1173148 9

[58] J. Ramon and T. Gartner. Expressivity versus efficiency of graph kernels. In *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, pp. 65–74, 2003. 4

[59] N. H. Riche, C. Hurter, N. Diakopoulos, and S. Carpendale. *Data-Driven Storytelling*. A K Peters/CRC Press, Boca Raton, FL, USA, 2018. 2

[60] L. E. Rocha, N. Masuda, and P. Holme. Sampling of temporal networks: Methods and biases. *Physical Review E*, 96(5):052302, 2017. 9

[61] G. Rossetti and R. Cazabet. Community discovery in dynamic networks: a survey. *ACM Computing Surveys (CSUR)*, 51(2):1–37, 2018. 9

[62] M. Ružička. Anwendung mathematisch–statistischer methoden in der geobotanik (synthetische bearbeitung von aufnahmen), 1958. 4

[63] B. Saket, P. Simonetto, and S. G. Kobourov. Group-level graph visualization taxonomy. *CoRR*, abs/1403.7421, 2014. doi: 10.48550/arXiv.1403.7421 2

[64] M. Saraceni. *The Language of Comics*. Psychology Press, 2003. 4, 5

[65] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, 2010. doi: 10.1109/TVCG.2010.179 1, 2, 3

[66] M. Shin, J. Kim, Y. Han, L. Xie, M. Whitelaw, B. C. Kwon, S. Ko, and N. Elmqvist. Roslingifier: Semi-automated storytelling for animated scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 29(6):2980–2995, 2023. doi: 10.1109/TVCG.2022.3146329 9

[67] L. South, D. Saffo, O. Vitek, C. Dunne, and M. A. Borkin. Effective use of likert scales in visualization evaluations: A systematic review. *Computer Graphics Forum*, 41(3):43–55, 2022. doi: 10.1111/cgf.14521 8

[68] C. D. Stolper, B. Lee, N. H. Riche, and J. Stasko. Emerging and recurring data-driven storytelling techniques: Analysis of a curated collection of recent stories. Technical Report MSR-TR-2016-14, Microsoft Research, 2016. 3

[69] S. Suh, J. Zhao, and E. Law. CodeToon: Story ideation, auto comic generation, and structure mapping for code-driven storytelling. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp. 13:1–13:16. ACM, New York, NY, USA, 2022. doi: 10.1145/3526113.3545617 2, 3

[70] C. Tong, R. C. Roberts, R. Borgo, S. P. Walton, R. S. Laramee, K. Wegba, A. Lu, Y. Wang, H. Qu, Q. Luo, and X. Ma. Storytelling and visualization: An extended survey. *Information*, 9(3):65, 2018. doi: 10.3390/info9030065 2, 3

[71] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Dynamic network visualization with extended massive sequence views. *IEEE Transactions on Visualization and Computer Graphics*, 20(8):1087–1099, 2014. doi: 10.1109/TVCG.2013.263 2

[72] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):1–10, 2015. doi: 10.1109/TVCG.2015.2468078 2, 3

[73] S. Varma, S. Shivam, A. Thumu, A. Bhushanam, and D. Sarkar. Jaccard based similarity index in graphs: A multi-hop approach. In *Proceedings of IEEE Delhi Section Conference*, pp. 1–4, 2022. 4

[74] C. Vehlow, F. Beck, and D. Weiskopf. Visualizing group structures in graphs: A survey. *Computer Graphics Forum*, 36(6):201–225, 2017. doi: 10.1111/cgf.12872 2, 3

[75] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner. Visual analysis of large graphs: state-of-the-art and future research challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011. doi: 10.1111/j.1467-8659.2011.01898.x 2, 3

[76] Q. Wang, Z. Li, S. Fu, W. Cui, and H. Qu. Narvis: Authoring narrative slideshows for introducing data visualization designs. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):779–788, 2019. doi: 10.1109/TVCG.2018.2865232 2

[77] Z. Wang, J. Ritchie, J. Zhou, F. Chevalier, and B. Bach. Data comics for reporting controlled user studies in human-computer interaction. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):967–977, 2021. doi: 10.1109/TVCG.2020.3030433 2

[78] Z. Wang, H. Romat, F. Chevalier, N. H. Riche, D. Murray-Rust, and B. Bach. Interactive data comics. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):944–954, 2022. doi: 10.1109/TVCG.2021.3114849 2, 3

[79] Z. Wang, S. Wang, M. Farinella, D. Murray-Rust, N. H. Riche, and B. Bach. Comparing effectiveness and engagement of data comics and infographics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 1–12. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300483 2

[80] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127, 2021. doi: 10.1109/TAI.2021.3076021 2

[81] J. Xu, Y. Tao, Y. Yan, and H. Lin. Exploring evolution of dynamic networks via diachronic node embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 26(7):2387–2402, 2018. doi: 10.1109/TVCG.2018.2887230 2, 3

[82] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, and X. Yang. A short survey of recent advances in graph matching. In *Proceedings of the International Conference on Multimedia Retrieval*, p. 167–174, 2016. 4

[83] V. Yoghourdjian, T. Dwyer, K. Klein, K. Marriott, and M. Wybrow. Graph thumbnails: Identifying and comparing multiple graphs at a glance. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3081–3095, 2018. doi: 10.1109/TVCG.2018.2790961 2

[84] J. Zhao, S. Xu, S. K. Chandrasegaran, C. Bryan, F. Du, A. Mishra, X. Qian, Y. Li, and K. Ma. ChartStory: Automated partitioning, layout, and captioning of charts into comic-style narratives. *IEEE Transactions on Visualization and Computer Graphics*, 29(2):1384–1399, 2023. doi: 10.1109/TVCG.2021.3114211 3, 8

[85] Z. Zhao, W. Benjamin, N. Elmqvist, and K. Ramani. Sketcholution: Interaction histories for sketching. *International Journal of Human-Computer Studies*, 82:11–20, 2015. doi: 10.1016/j.ijhcs.2015.04.003 3

[86] Z. Zhao, R. Marr, and N. Elmqvist. Data comics: Sequential art for data-driven storytelling. Technical Report 15, Human-Computer Interaction Laboratory, University of Maryland, College Park, 2015. 1, 2, 3